

*In the Claims:*

Please amend claims 27-48, 71-75, 78, 79, and 81 as follows. Claims 49, 76, 77, 82, and 83 remain unchanged. Please add the following new claims 84-89.

27. (Twice Amended) The method of claim 71 wherein the client library determines the server application based on an association with the class identifier.

28. (Twice Amended) The method of claim 27 wherein the association is recorded during installation of the server application.

29. (Twice Amended) The method of claim 27 wherein the association is recorded when the server application is launched.

30. (Once Amended) The method of claim 27 including when the server application supports an object that is compatible with the client application, launching the server application.

31. (Once Amended) The method of claim 30 wherein the client application is executing in a process and the server application is launched in a separate process.

32. (Once Amended) The method of claim 30 wherein the client application is executing in a process and the server application is launched in the same process.

33. (Once Amended) The method of claim 30 wherein the client application and the server application exchange data using a compatible format.

34. (Twice Amended) The method of claim 27 wherein the client library determines the association while the server application is not executing.

35. (Twice Amended) The method of claim 71 wherein the server application records in the configuration store, an association between itself and the class identifier.

36. (Once Amended) The method of claim 35 including when the server application supports a data format that is compatible with the client application, launching the server application.

37. (Once Amended) The method of claim 36 wherein the client application is executing in a process and the server application is launched in a separate process.

38. (Once Amended) The method of claim 36 wherein the client application is executing in a process and the server application is launched in the same process.

39. (Once Amended) The method of claim 36 wherein the client application and the server application exchange data using a compatible format.

40. (Twice Amended) The method of claim 71 wherein the client application, the client library, the server application, and the server library, each execute in a separate process.

41. (Twice Amended) The method of claim 71 for supplying the server application to perform the requested manipulation wherein the server application populates the configuration store with class identifiers it supports.

42. (Twice Amended) The method of claim 41 wherein the client application determines the server application that supports the class identifier while the server application is not executing.

43. (Twice Amended) The method of claim 41 wherein the server application populates the configuration store during installation of the server application.

44. (Twice Amended) The method of claim 41 wherein the server application populates the configuration store when the server application is launched.

45. (Twice Amended) The method of claim 41 including when the server application supports a data format that is compatible with the client application, launching the server application.

46. (Once Amended) The method of claim 45 wherein the client application is executing in a process and the server application is launched in a separate process.

47. (Once Amended) The method of claim 45 wherein the client application is executing in a process and the server application is launched in the same process.

48. (Once Amended) The method of claim 45 wherein the client application and the server application exchange data using a compatible format.

49. (Twice Amended) A computer-readable medium containing instructions for causing a computer system to perform the method of claim 71.

71. (Twice Amended) A method in a computer system for manipulating an object displayable in a client application via any one of a plurality of server applications using an application programming interface, the computer system having a configuration store for storing a class identifier associated with the object and associating the class identifier with at least one of the server applications of the plurality of server applications, the method comprising:

requesting by the client application through the application programming interface, a manipulation to be performed on the object wherein routines of the application programming interface are divided into an object-independent client library and a server library, the object-independent client library comprising routines which invoke the proper server application to manipulate the object, and the server library comprising routines which process requests to manipulate the object;

determining by the object-independent client library using the configuration store and the class identifier of the object, a server application out of a plurality of server applications to perform the requested manipulation on the object;

1  
sending by the object-independent client library, a message to the server library to perform the requested manipulation on the object;  
receiving by the server library the message to perform the requested manipulation on the object; and  
invoking by the server library the server application to perform the requested manipulation on the object.

---

~~DH~~  
72. (Once Amended) The method of claim 71, wherein the object displayable in the client application is a first object, the method further comprising:  
depicting the first object as appearing inside a second object displayable in the client application.

73. (Once Amended) The method of claim 71, wherein the client application determines from the configuration store and displays for a user a list of available manipulations on the object.

74. (Once Amended) The method of claim 71, wherein the server application is started up in response to receiving the message.

75. (Once Amended) The method of claim 71, wherein the server application shuts down after completion of manipulations requested in the message.

---

~~DH~~  
76. (Unchanged) The method of claim 71 wherein a user can select a new object from amongst a plurality of embedded or linked objects displayed in a graphical user interface.

77. (Unchanged) The method of claim 71 wherein a user can select a manipulation or procedure to be performed on a selected object from amongst a plurality of manipulations or procedures displayed in a graphical user interface.

DIS 78. (Once Amended) A method in a computer system, the computer system having a configuration store for storing identifiers of available embedded or linked objects and identifiers of servers associated with the embedded or linked objects, the method comprising:

providing an application programming interface supporting embedded or linked objects wherein the application programming interface is separated into an object-independent client library and a server library, the object-independent client library comprising routines which determine servers to manipulate embedded or linked objects, and the server library comprising routines which invoke server routines on embedded or linked objects;

requesting by a user from a client application, creation of an embedded or linked object;

requesting by the client application via the application programming interface to the object-independent client library, available embedded or linked objects;

determining by the object-independent client library from the configuration store and returning to the client application, a list of the available linked or embedded objects;

presenting by the client application to the user, the available embedded or linked objects;

selecting by the user from the available presented objects an object to be linked or embedded within a container object displayed via the client application;

requesting by the client application via the application programming interface to the object-independent client library, creation of the selected embedded or linked object;

determining by the object-independent client library from the configuration store a server associated with the selected linked or embedded object and sending a message from the object-independent client library to the server library, to create the selected linked or embedded object;

receiving by the server library, the message to create the selected embedded or linked object; and

invoking by the server library at the determined server, a server routine to create the selected embedded or linked object;

whereby the created linked or embedded object is created by the server routine and the user can edit or otherwise manipulate the created linked or embedded object within the container object.

1) global "registry." This registry is a database of information such as (1) for each type of object, the server application that implements the object type, (2) the actions that each server application provides to client applications, (3) where the executable files for each server application are located, and (4) whether each server application has an associated object handler.

---

Please replace the paragraph at page 11, line 30, with the following paragraph:

---

D4 Communication between client and server processes occurs either synchronously or asynchronously. Synchronous communication occurs when one process sends a message to the other process and the sending process suspends activity until the other process completely processes the message. For example, when a client process wants to create a new containee object, the client process (through the client library routines) sends a message to the appropriate server process. The client process waits until the containee object is created before continuing execution. Asynchronous communication occurs when one process sends a message to the other process and the sending process continues execution while the receiving process responds to the message. Typically, when the receiving process has completed responding to the request, it sends a message indicating such completion to the sending process. For example, when a client process wants a containee object saved in a compound document file, the client process sends a message to the server process. The client process can continue to execute (e.g. responding to users requests) while the server process is saving the object. When the server process has completed saving the object, it sends a completion message to the client process.

---

Please replace the paragraph at page 35, line 13, with the following paragraph:

---

DS Although the present invention has been described in terms of a preferred embodiment, it is not intended that the invention be limited to this embodiment. Modifications within the spirit of the invention will be apparent to those skilled in the art. The scope of the present invention is defined by the claims which follow.

79. (Once Amended) The method of claim 78 wherein the user is able to edit or manipulate a linked or embedded object by selecting an action available on a client application menu.

81. (Once Amended) A computer software system comprising:  
client means for displaying containee objects within the client means, wherein containee objects are associated with object class identifiers;  
plural server means for performing manipulations on containee objects;  
configuration store means for storing an association between object class identifiers and server means that perform manipulations on associated object classes;  
client library means for receiving from the client means, requests to perform manipulations on containee objects, consulting the configuration store means and determining an appropriate server means out of the plural server means to perform manipulations based on object class identifiers, and sending messages to server library means associated with determined server means, said messages comprising requests to perform manipulations on containee objects;  
and  
server library means for receiving messages to perform requested manipulations, and invoking server routines for performing requested manipulations.

D16 82. (Unchanged) A computer-readable medium containing instructions for causing a computer system to perform the method of claim 27.

83. (Unchanged) A computer-readable medium containing instructions for causing a computer to perform the method of claim 78.

D17 84. (New) The method of claim 71 further comprising:  
receiving by the server library, an indication from the server application that the requested manipulation is complete;  
sending a message from the server library that the requested manipulation is complete;  
receiving by the client library, the message from the server library that the requested manipulation is complete; and

sending an indication to the client application that the requested manipulation is complete.

85. (New) The method of claim 71 wherein the client application and the client library are dynamically linked to execute in the same process.

86. (New) The method of claim 71 wherein the server application and the server library are dynamically linked to execute in the same process.

87. (New) The method of claim 71 wherein the client library and the server library send messages via a channel comprising inter-process communication.

88. (New) The method of claim 71 wherein the client application, the client library, the server application, and the server library are processes sharing the same processor.

89. (New) The method of claim 71 wherein the application programming interface provides functions comprising compound document functionality.

---